

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

A Multi Objective Evolutionary Algorithm for Solving a Real Health Care Fleet Optimization Problem

Carlos Catania^{a,*}, Cecilia Zanni-Merk^{a,b}, François de Bertrand de Beuvron^{a,b}, Pierre Collet^a

^aICube laboratory, BFO team, Strasbourg University, CNRS, France

^bICube laboratory, BFO team, INSA de Strasbourg, CNRS, France

Abstract

The problem of the transportation of patients from or to some health care center given a number of vehicles of different kinds can be considered as a common Vehicle Routing Problem (VPR). However, in our particular case, the logistics behind the generation of the vehicle itineraries are affected by a high number of requirements and constraints such as the enterprise benefits, the satisfaction of the patients, and the respect of certain law regulations regarding the patients and the employees. In this work, we discuss the main aspects of the implementation of a Multi Objective Evolutionary Algorithm focused on providing a set of valid solutions to the end users of Patient Transport Services. We provide a detailed description of the process of integrating all the information on different genetic operators and multiple fitness functions. Finally, we present the preliminary results on a real-life problem from an small company that provides transport service and we compare the results that our implementation gets with the itineraries proposed by human experts.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of KES International

Keywords: multi-objective optimisation, genetic algorithm, hard and soft constraints

1. Introduction

In several regions of France, Patient Transportation Service (PTS) companies are in charge of the transportation of patients from or to some health care centre. This task usually includes dealing with a given a number of vehicles and considering several requirements and constraints. The logistics behind the generation of the vehicle itineraries are affected by company resources such as the number of employees, vehicles, the different patients requests, among other requirements. The fact is that until now, companies providing PTS used to deal with the generation of vehicle itineraries by means of the experience of human operators. Vehicles itineraries were generated manually in order to guarantee (and maximise) the company benefits, but also the Quality Of Service (QOS) and the satisfaction of certain

* Corresponding author. Tel.: +54-261-592-9560

E-mail address: catania@unistra.fr

law regulations. But the problem is that these itineraries can soon become unrealistic after a certain time of the day, due to unplanned trips that were needed to be dealt with.

Given the amount of requirements and constraints observed, it soon became clear that the application of a Multi Objective Evolutionary Algorithm (MOEA)^{1,2} could be appropriate to deal with the whole problem. The use of approaches based on MOEA provides the user with a precise mechanism for organizing all the requirements and constraints.

In the present article, we describe the procedure for solving the generation of the PTS itineraries by means of a MOEA. In particular we focus on how the genetic operators and objective functions can deal with the constraints and requirements of the problem and we make a preliminary evaluation of its performance on a real-life situation.

The MOEA is in fact the heart of a larger application which includes more traditional management modules such as customer billing, customer support, calculation of working hours and payment of ambulance attendants according to the legislation in force, among others. Therefore, the development of the different operators of the MOEA are in a direct relation with the classes and data structures inside the management application.

Even if this problem belongs to the family of Vehicle Routing Problems³ (VRP), it is clear that the solutions proposed in the literature, as specific as they are^{4,5,6,7}, were not intended to take into account all the specific constraints of a real problem. Even some very efficient genetic algorithms for the pick-up and delivery problem⁸ have not considered all the important details involved in the health care transport area. We believe that this situation is found in many optimization problems when the parameters are numerous and varied in nature. Thus, in our example, legal constraints such as the working time of ambulance attendants, or medical constraints such as the disinfection of the vehicles, or the personnel qualification, cannot be overlooked while minimizing costs or distances. The taking into account of all these constraints is the main contribution of our work.

The rest of the paper is structured as follows. In Section 2, we provide a detailed description of the elements of problem, a simple classification of the different kinds of constraints and the main goals to achieve during optimization. Then, in sections 3 and 4, we describe the objectives functions proposed to guide the evolutionary process considering the requirements and constraints. In section 5, we provide a description of the main aspects of the chromosome and the genetic operators. An evaluation of the algorithm with data corresponding to five days of real planning is conducted in sections 6 and 7. Finally, in section 8, we give the conclusions and the possible improvement in the development of this MOEA.

2. Description of the Problem

Let be $C = \{0, 1, \dots, c\}$ the set of customers who need transportation and $H = \{0, 1, \dots, h\}$ the set of health centers. We have a complete direct graph $G = (P, A)$, where $P = \{0, 1, \dots, v\}$ are the different addresses of the customers and health centers and $A = \{(i, j), i, j \in V, i \neq j\}$ is the set of arcs connecting the addresses. For each arc $(i, j) \in A$ is associated a benefit b_{ij} and a travel time $t_{ij}(\geq 0)$ and a distance $d_{ij}(\geq 0)$. $V = \{0, 1, \dots, v\}$ is the set of vehicles and $E = \{0, 1, \dots, e\}$ is the set of employees required to operate each vehicle. Each vehicle $k \in V$ has a capacity $Q \leq \max_q$ for transporting. Each employee r has a maximal number W of working hours and a mandatory pause every P hours. Each customer $l \in C$ has specific demands regarding the appointment time S_i at some given address i_h , a service duration SD_i and the required type of vehicle v_l . The problem is to design a set of routes $R = \{r_1, \dots, r_k\}$ such that each route begins at ends at health centers H and each customer is transported by exactly one vehicle. Routing design is performed considering several constraints/requirements. An enumeration of such constraints is shown in Table 1 categorized by the customer personal needs, the availability of the employees and the vehicle fleet. It is important to mention these are just a subset of the complete set of constraints observed in the problem.

2.1. Constraint Classification

When considering all the requirements that should be met for generating the itineraries, it comes out that it is possible to differentiate between *soft* and *hard constraints*. Soft constraints are requirements that are desirable to meet, whereas hard constraints are requirements that must be met in order to obtain a viable solution. As an example of a soft constraint we can mention the limit in the working hours of certain employees. The reality is that there will be situations in which the working hours limit could be exceeded when transporting a patient to hospital. In those cases, there will be no other option than ignoring the limit and finishing the trip to the hospital and assuming the

Table 1: Problem Constraints associated to the different entities

Employee	Customer	Vehicle
- Number of Employees $ E $ available.	- Required Type of Vehicle $e_l \in E$	- Number of Vehicles available $ V $.
- Max. W number of hours for Employee E_i .	- Respect the appointment Time S_i .	- Number of E for each Vehicle $v_i \in V$.
- Mandatory break every P hours.	- Possibility of sharing the Vehicle	- Max. number q of Passengers per vehicle v_i
	- Specificities: Wheeling chair, oxygen tank, etc.	- Max. number of Km d before next mechanical service for vehicle v_i .
	- Preferred Crew.	- Vehicle Disinfection required.

associated cost. Even if a constraint has been violated, the trip is still possible. On the other hand, the maximum occupancy capacity of a vehicle is a physical limitation that cannot be exceeded. In other words, an ambulance that is prepared to receive just one patient, cannot, under any circumstances, deal with two or more patients. This latter case is a hard constraint. Nevertheless, it is important to mention that the PTS company will ultimately be responsible of differentiating between hard and soft constraints.

2.2. Optimization Axes

In addition to the constraint satisfaction, the PTS company aims at optimising the itineraries considering the following axes:

Economical , including all the requirements and constraints that have some associated monetary cost or income. For instance, the number of kilometer associated with each vehicle or the number of working hours of each employee.

Quality of Service , including all the requirements and constraints related to patients. For instance, respecting the appointment time for a given patient.

Physical , including all the requirements and constraints related to physical aspects of the problem. For instance, the maximum number of passengers a given vehicle can transport at the same time.

These three axes will provide the basis for the specification of the objective functions required for the MOEA that will be described in the following sections.

2.3. Problem Element description

Given the huge quantity of details involved in the problem, we proceeded to formalize it as a domain ontology⁹. By using such an ontology, the formal relationships that appear among the entities help us in the definition of the MOEA elements on this particular problem. Such ontology was in direct relation with the classes and data structures inside the management application. A description of the methodology followed for building the domain ontology was published in¹⁰. We describe the main entities necessary for describing EA operators.

PlannedElement is one of the essential components of the routing problem. It is an event to be held at a given location P_{th} (Address) which should start at a service time S_i (requestedDate), and should last for a known or estimated time length SD_i (duration). There are many subtypes of *PlannedElement* such as the *PlannedEmployeeElement* or the *PlannedVehicleElement*. The first refers to events related to employees such as the mandatory break P_e every 6 hours. The second, refers to events related to vehicles, such as when a vehicle must be disinfected or mechanically inspected.

PlannedBusinessFleetElement is a subtype of *PlannedElement* that describes an event where a patient should be **picked-up** or **delivered** to some health care center. It is the most frequent event.

Journey is used for representing the relation between two *PlannedBussinesFleetElement*. This element is used for representing the obvious fact that a patient that was picked-up by a vehicle must be delivered to the hospital by the same vehicle.

PlanningLine represents the list, ordered by increasing time, of the events of type *PlannedElement* supported by a given vehicle. As a result, each vehicle v_i will have associated an ordered list $P_i = [p_1^i, \dots, p_{n_i}^i]$ of *PlannedElements* assigned to the vehicle.

Planning represents the set of *PlanningLines* for all the vehicles. A *Planning* is the complete set of vehicle itineraries for a given set of Journeys.

3. Multi Objective Optimization

A multi-objective optimization problem can be described in mathematical terms as follows:

$$\min F(x) = \min [f_1(x), f_2(x), \dots, f_n(x)], x \in S,$$

where $n > 1$ and S is the set of constraints as defined above. The space in which the objective vector belongs is called the *objective space*, and the image of the feasible set under F is called the *attained set*. Such a set will be denoted in the following with $C = \{y \in R^n : y = F(x), x \in S\}^1$.

The scalar concept of “optimality” does not apply directly in the multi-objective framework. Here the notion of *Pareto optimality* has to be introduced. Essentially, a vector $x^* \in S$ is said to be *Pareto optimal* for a multi-objective problem if all other vectors $x \in S$ have a higher value for at least one of the objective functions f_i , with $i = 1, \dots, n$, or have the same value for all the objective functions.

A point x^* is said to be a *Pareto optimum* or a *efficient solution* for the multi-objective problem if and only if there is no $x \in S$ such that $f_i(x) < f_i(x^*)$ for all $i \in 1, \dots, n$.

The image of the efficient set, i.e., the image of all the efficient solutions, is called *Pareto front* or *Pareto curve* or *surface*. The shape of the Pareto surface indicates the nature of the trade-off between the different objective functions.

For dealing with the generation of vehicle itineraries, we have chosen the NSGA-II¹¹ algorithm. The NSGA-II is a well known MOEA, which efficiently supports the multi-objective approach. It has been proved that NSGA-II is able to find a better spread of solutions with better convergence near the true Pareto-optimal front.

4. Objective functions

The different kinds of constraints and requirements can be grouped in different objective functions by using the classification provided in section 2. For instance, the economical axis arises as a clear objective function, in which all constraints and requirements regarding vehicles and employees costs can be included. Similarly, the QoS axis includes all the information about the service and the patient.

On the other hand, the so called hard constraints, such as the maximum number of occupants, are modelled as an objective function and also included in the multi-objective algorithm. A similar approach was described in¹². Here, the main idea is to have $m + n$ objectives, where n is the number of constraints and m the number of actual objectives. Then, we can apply a multi-objective optimization technique to the vector $v = (f, f_1, \dots, f_n, f_{n+1}, \dots, f_m)$ where $f_{n+1} \dots f_m$ are the constraints. Then, a valid solution X would have $f_i(X) = 0$ for $m + 1 \leq i \leq n$.

We find that by including hard constraints as part of the fitness functions, we can easily include newer hard constraints to the algorithm without producing major modifications to the chromosome. At the end of the evolution process, only those individuals that have satisfied these constraints (e.g. considered as valid individuals) will be available for the vehicle fleet management application. A detailed description of the implemented fitness functions is shown below.

4.1. Calculating the Economical axis

The economical axis is one of the most important aspects to be considered when calculating an itinerary. In the current implementation we have found that the calculation of the benefits arises as the most natural approach to deal with most of the economical details of the problem. Most of the fine economical details related to a itinerary can be included in a fitness function, which considers on the one hand the Billings and on the other hand the associated Cost.

In general, computing the billing for a *Journey* j , includes the distance d between address of the *PlannedElements* p_j and p_{j+1} times a price per kilometer (P_{km}). The billing price can be increased when the number of traveled kilometers is below specific ranges. On the other hand, the billing price is decreased if a particular customer can share the vehicle with someone else. In this way, the final function for calculating the billing of a given journey j can be expressed as $Billing(j) = (d_{p_j, p_{j+1}} P_{km} + I(d_{p_j, p_{j+1}}), CNO_j)$, where $I(d_{p_j, p_{j+1}})$ refers to the increase suffered when the specific number of kilometers is traveled and CNO_j refers to the concurrent number of occupants.

The distance $d_{p_j, p_{j+1}}$ is also considered for calculating the cost associated to *Journey* j . In this case, we have to consider the cost per kilometer of the given vehicle CKm_v . In addition, we need to calculate the cost for the *Crew* related to the vehicle. The cost of a *Crew* will depend on the number of employees and the cost per hour CHs_e of each one of them. We can express the cost function more formally as:

$$Cost(j) = d_{p_j, p_{j+1}} CKm + \sum_{e \in Crew} CHs_e$$

Finally, the objective function can be calculated as:

$$\max \sum_{v \in V} \sum_{p \in P_v} Billing(inJ(p)) - Cost(inJ(p))$$

where P_v is the list of *PlannedElements* corresponding to vehicle v and $inJ(p)$ denotes the set containing p and all the *PlannedElements* in the same *Journey* as p .

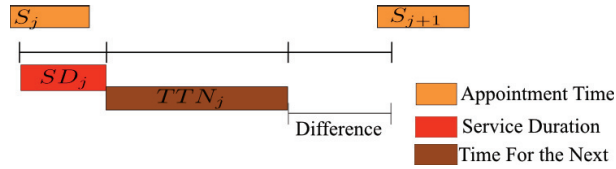


Fig. 1: Temporal line between points p_j and p_{j+1}

4.2. Calculating the Quality of Service

In the current version of the MOEA, we have focused on the cost associated with being late or early. The idea is to minimize the waiting time of the patient (in the case of being late) as well of the waiting time of the vehicle (in the case of being early).

Each *Journey* J between two *PlannedElements* p_j and p_{j+1} is represented as a temporal line (cf. Fig. 1) with three values: S_j , SD_j and (TTN_j) . S_j refers to the time the vehicle should arrive at destination. SD_j is the time needed to take care of the patient and TTN_j is the estimated time to arrive to destination.

For the current implementation, we estimate the SD_j value in 15 minutes, whereas information concerning the TTN_j is pre-calculated using a third-party software and stored as a three-dimensional matrix (called *cubeTTN*), where each element represents the travel times between two addresses for a given (discrete) time in the day. Therefore, $cubeTTN_{ijt}$ is the time in seconds needed to go from location i to location j at time of da day td .

The cost is represented through a piecewise function $f_0(x)$, where $x = S_{j+1} - (S_j + SD_j + TTN_j)$ is the temporal difference shown in Fig. 1. If this difference is negative (late arrival), the cost is quadratic; otherwise (early arrival), the cost is linear.

$$f_0(x) = \begin{cases} x^2, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Therefore, the objective can be formalized as:

$$\min \sum_{v \in V} \sum_{p \in P_i} f_0(S_{j+1} - (S_j + S D_j + \text{cubeTTN}(p_j, p_{j+1}, S_{j+1})))$$

4.3. Calculating the Physical Axis

In the current implementation of the algorithm we deal with the maximum number of occupants that a given vehicle (denoted as maxOccup_v) can transport at the same time. The approach is represented through the piecewise function $\text{offset}(x)$, where x is the current number of occupants for a given PE . If x is greater than maxOccup_v , the offset is calculated considering the absolute value of the difference maxOccup_v and x . If x is less than zero, then the offset is calculated in the opposite way ($x - \text{maxOccup}_v$). Finally, if x is less than maxOccup_v , 0 is returned.

$$\text{offset}(x) = \begin{cases} \text{abs}(\text{maxOccup}_v - x), & x > \text{maxOccup}_v \\ \text{abs}(x - \text{maxOccup}_v), & x < 0 \\ 0, & x < \text{maxOccup}_v \end{cases}$$

Therefore, the objective can be formalized as:

$$\min \sum_{v \in V} \sum_{p \in P_i} \text{offset}(\text{occupant}_p)$$

5. Definition of the Chromosome and Variation Operators

The structure of the chromosome is presented in Fig. 2. As it can be seen, the chromosome represents a complete Planning where all the vehicles and all the *PlannedElements* are considered. For every available vehicle, we associate a crew and the list of points (*PlannedElements*) that the vehicle needs to attend. The list of *PlannedElements* defines the *PlanningLine* associated to each vehicle. As mentioned, the *PlanningLine* is sorted in ascending order according to the requested time for the event.

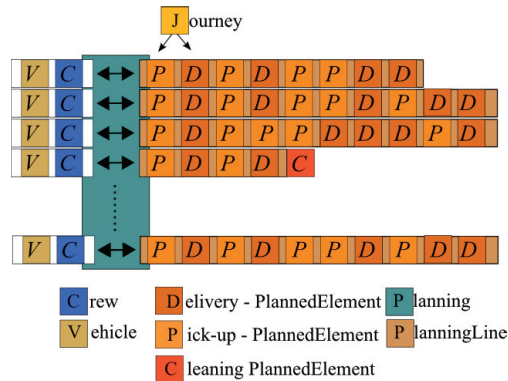


Fig. 2: Structure of the chromosome

For initializing each chromosome, a predefined number of vehicles are uniformly and randomly selected from the complete set of available vehicles. For each vehicle, a random Crew is uniformly and randomly selected, respecting the constraints about the number of employees per vehicle. Then, in a second stage, journeys are added to the corresponding *PlanningLine* respecting the required vehicle type constraint. *PlannedElements* belonging to a *Journey* are inserted ordered by the appointment date. If the customer request is not to share the vehicle, *PlannedElements* belonging to that *Journey* should be inserted contiguously in the *PlanningLine* (see the first two *PlannedElements*

in the first *PlanningLine* of Fig. 2. In the opposite situation, both *PlannedElements* can be distant allowing the inclusion of new Journeys between them (see the last four *PlannedElement* in the first *PlanningLine* of Fig. 2). For the case of Vehicle disinfection, a third *PlannedElement* is created and added following the corresponding Delivery *PlannedElement* in the *PlanningLine*.

5.1. Variation operators

These operators should take into account all the requirements and constraints that are necessary for the planning to tend towards a possible solution. There are two major axes that differ in the structure of a chromosome: the distribution of the *Crews* and the *PlannedElements* on the set of vehicles. Any changes in each one of the two axes will affect the values associated to the objective functions and will provide the means for evolving populations with individuals of better quality. Several operators for the two axes have been defined, including Crossover, Mutation and Swap. For space reason just two of them are described below.

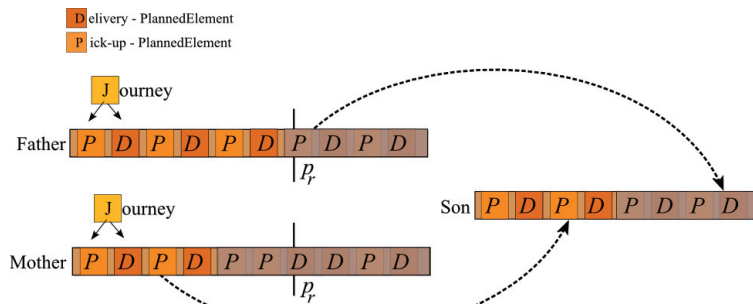


Fig. 3: *PlannedElement* Crossover operator

5.1.1. *PlannedElement* Crossover

A graphical description of the crossover operator is shown in Fig. 3. In general terms, the crossover operator is implemented following a classical one point crossover approach.

As it was mentioned *PlannedElements* are grouped in a *Journey* and therefore should be assigned to a same vehicle. A situation that is considered by the operator. To perform the crossover between two individuals, named mother and father in the figure, a *PlannedElement* p_r is randomly chosen in this list. For each *PlanningLine* i , we have the list P_i^f (resp. P_i^m) of *PlannedElements* affected to vehicle i in the father (resp. mother) individual. The set P_i^s of *PlannedElement* affected to vehicle i in the new son individual is defined by :

$$[I]P_i^s = \{p^f \in P_i^f; \exists p \in \text{inJ}(p^f) S_p \leq S_{p_r}\} \cup \{p^m \in P_i^m; \forall p \in \text{inJ}(p^m) S_p > S_{p_r}\}$$

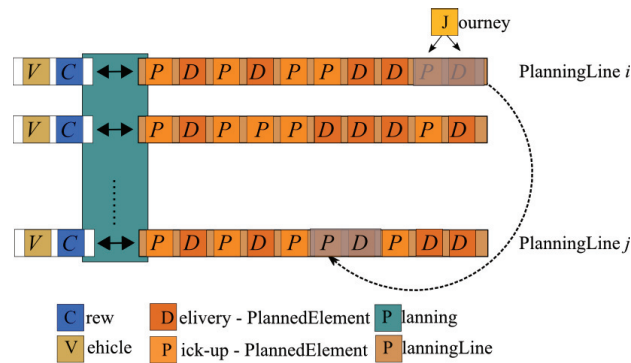
In other words, the son gets assigned the earliest *PlannedElements* of its father and the latest ones of its mother.

5.1.2. *PlannedElement* Mutation

Similarly to the *PlannedElement* crossover operator, we have to deal with the constraint regarding those *PlannedElement* belonging to a *Journey*. A graphical representation of the mutation operator is illustrated in Fig. 4. To perform the mutation on one previously selected individual, we randomly select a *PlanningLine* i and a *PlannedElement* p_r . Then, all the *PlannedElements* that belong to the *Journey* (e.g. $\text{inJ}(p_r)$) are removed from *PlanningLine* i and re-inserted in a *PlanningLine* j . It is important to mention that the insertion in the *PlanningLine* j is done maintaining the proper temporal order.

5.2. Pause Operator

A special kind of operator was added to deal with the mandatory break every 6 hours every employee should take. The Pause operator simply inserts a *PlannedEmployeeElement* after calculating the number of hours a employee has been working. The insertion is done in a moment when no patients are on the vehicle. In opposition to the variation operators, the Pause Operator is executed with a probability of 1.

Fig. 4: *PlannedElement* Mutation operator

6. Experiment Setup

A first prototype of the algorithm was developed using EASEA^{13,14} a high-level framework for easily developing evolutionary approaches. The framework was extended to support multi-objectives algorithms by including the source code of the NSGA-II¹¹ algorithm.

In order to evaluate the behaviour of the algorithm, we carried out an experiment on an average sized problem. Real data is coming from a small PTS company with a 23 vehicle fleet. The vehicle fleet is composed of VSL (e.g. the French initials for Lightweight Sanitary Vehicle) and Ambulances. For this paper, we are using the data corresponding to the vehicle itineraries for five days (from Monday to Friday).

The three first columns of Table 2 show the number of journeys and vehicles used for satisfying the customer's request for each day. The vehicle itineraries for each day as they were planned by the PTS staff have been evaluated using the objective functions described in section 3. The results of that evaluation are shown in columns four (Delay function from QoS Axis) and five (Benefits Function from Economical Axis) of Table 2. Given that the result of the evaluation of the objective function from the physical axis of section 4.3 is zero, we exclude it from the table. Notice that the negative values for the Benefits function indicates losses for each one of the evaluated days. A situation that was confirmed by the PTS staff.

Table 2: Evaluation in terms of the proposed fitness functions for the five days of real planning provided by the PTS company

Day of the week	Number of Journeys	Number of Vehicles	Fitness Delay	Fitness Benefits
Monday	159	21	1.37E-04	-4.83E+07
Tuesday	132	21	8.28E-05	-1.32E+07
Wednesday	126	21	4.85E-05	-1.41E+07
Thursday	146	23	4.86E-05	-2.01E+07
Friday	115	21	1.17E-4	-1.15E+07

The evolutionary algorithm is executed 35 times for each day using a probability of 0.4 for the crossover operator and 0.1 for the mutation operators.

7. Experiment Results

In the Table 3, we present the average and standard deviation (σ) of the fitness values obtained by the MOEA on the five days of data provided by the PTS company. As stated in section 4, only solutions satisfying the vehicle maximum number of occupants are considered for calculating the average values. That is, we consider only those solution whose values for the objective function from the physical axis of section 4.3 equals to zero.

As it can be seen, the average values for the delay and benefits objective functions are considerable better than the ones observed in Table 2. Moreover, in some cases the values show a difference of one order of magnitude. This is the case of the first 4 days in the Benefits fitness function and the first and the fifth days in the case of the Delay fitness function.

Table 3: Average fitness values and standard deviations for the five days of real planning provided after 35 executions of the MOEA

Day of the week	Fitness Delay	σ	Fitness Benefits	σ
Monday	2.40e-05	1.21e-05	-6.45e+06	2.40e+06
Tuesday	9.85e-06	9.48e-06	-6.87e+06	1.91e+06
Wednesday	1.25e-05	1.36e-05	-4.94e+06	1.54e+06
Thursday	1.06e-05	7.45e-06	-5.57e+06	2.75e+06
Friday	6.23e-06	9.00e-06	-5.91e+06	2.15e+06

A graphical representation of the solutions for the five days of data is shown in the Box plot from Figure 5. Regarding the Delay fitness function, we can observe the planning provided by the PTS company for Monday, Tuesday and Friday shows a lower performance than the worst solution found by the MOEMA. In the two other cases the solution provided by PTS are closer to the upper limit of the error box, but performing considerable worse than the solution included in the IQR (interquartile range). In the case of the Benefits function, the solutions provided by the PTS company presents bigger losses than the worst solution provided by the MOEA represented by the upper whisker of the box plot.

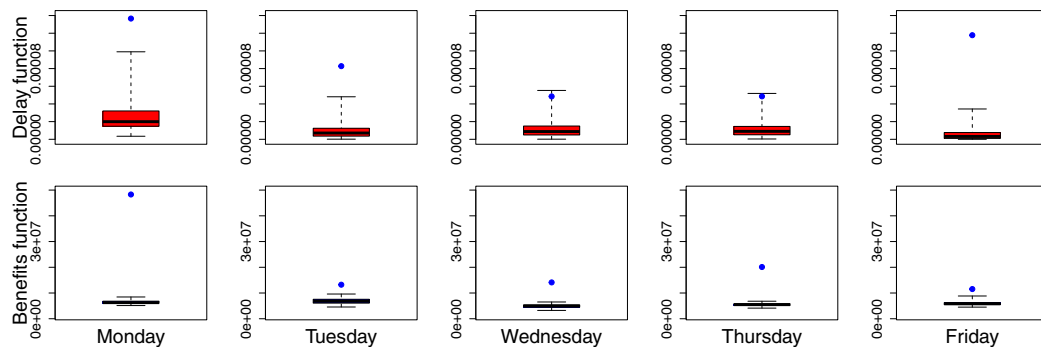


Fig. 5: Comparison between the Quality of the solutions provided by the PTS staff and all the solution found by the MOEA. Benefits and Delay functions are considered.

In Figure 6 we provide an overview of all the solutions provided by a particular execution of the MOEA. The figure shows all the solutions that have satisfied the maximum number of occupants for the five days of data (as described in section 4). The plot shows the different values for the Benefits and Delay functions in black point and the Pareto front found for each solution is shown in red. As can be seen, the Pareto front found by the MOEA has been able to provide valid solutions with different trade-offs between the Benefits and Delay fitness functions.

8. Summary and Conclusion

Even when the use of MOEA on VRP problems have been deeply analysed in the past years, the fact is that there are not many examples of real situations where it is necessary to deal with a high number of constraints and requirements. The modelling of problem through a domain ontology has provided us with an efficient way of dealing with the codification process of the different genetic operators.

By developing a multi-objective genetic algorithm, we have been able to include most of the constraints (hard and soft) into different fitness functions. In particular, the use of a fitness function for dealing with the physical constraints of the maximum number of occupants per vehicle has proven to be an efficient mechanism for dealing with hard constraints without major modifications in the algorithm.

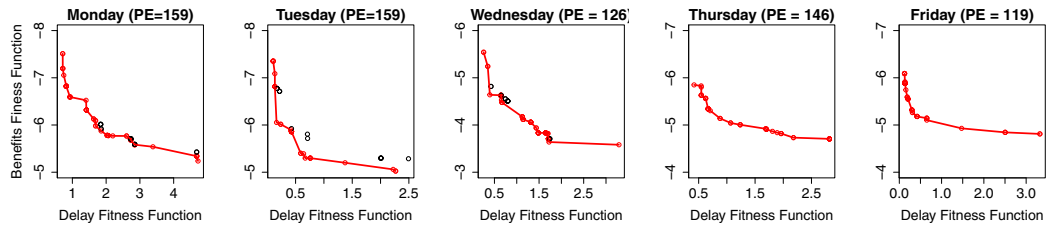


Fig. 6: Plot of the Pareto Front found in one execution of the MOEA for five days of data

The viability of the algorithm has been tested using five days of data coming from a real PTS company. The result has shown that the algorithm outperforms in all the solutions originally proposed by the PTS staff, while satisfying the constraint and requirements of the problem. In addition the MOEA has provided to the end-user a set of valid solutions considering different values for the Economical and QoS axis.

The evaluation of the current implementation of the MOEA has provided us with useful information for future improvements. For instance, the number of available vehicles is currently prefixed by the operator. It would be interesting to let the MOEA to select the optimal number of vehicles necessary to satisfy the customer requests for a given day.

The final goal of the algorithm is to be capable of providing a solution in less than a minute, which will allow it to deal with unplanned trip such as emergencies or some other urgent requirements. This is a goal that we expect to reach by means of the parallelization of the algorithm on GPGPU platforms taking advantage of the EASEA framework capabilities.

References

1. A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation* 1 (1) (2011) 32 – 49.
2. R. Lahyani, M. Khemakhem, F. Semet, Rich vehicle routing problems: From a taxonomy to a definition, *European Journal of Operational Research* 241 (1) (2015) 1 – 14.
3. B. Golden, R. Raghavan, E. Wasil (Eds.), *The Vehicle Routing Problem : Latest Advances and New Challenges*, Vol. 43 of Operations Research/Computer Science Interfaces Series., Springer, 2008.
4. J. Berger, M. Barkaoui, A parallel hybrid genetic algorithm for the VRP with time windows, *Computers & Operations Research* 31 (12) (2004) 2037 – 2053.
5. O. Bräysy, M. Gendreau, Vehicle routing problem with time windows, Part II: Metaheuristics, *Transportation Science* 39 (1) (2005) 119–139.
6. N. Carrasquero, J. A. Moreno, A new genetic operator for the travelling salesman problem, in: *Proceedings of the 6th Ibero-American Conference on AI: Progress in Artificial Intelligence, IBERAMIA '98*, Springer-Verlag, London, UK, UK, 1998, pp. 315–325.
7. A. Garcia-Najera, J. A. Bullinaria, An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows, *Computers & Operations Research* 38 (1) (2011) 287 – 300, project Management and Scheduling.
8. G. Pankratz, A grouping genetic algorithm for the pickup and delivery problem with time windows, *OR Spectrum* 27 (1).
9. S. Staab, R. Studer (Eds.), *Handbook on Ontologies*, International Handbooks on Information Systems, Springer, 2004.
10. C. C. François de Bertrand de Beuvron, Cecilia Zanni-Merk, Knowledge intensive evolutionary algorithms, in: *Proceedings of the 7th International Conference on Information, Process and Knowledge Management (eKNOW2015)*, 2015.
11. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *Trans. Evol. Comp* 6 (2) (2002) 182–197.
12. E. Mezura-Montes, C. A. C. Coello, A survey of constraint-handling techniques based on evolutionary multiobjective optimization, in: *Workshop paper at PPSN*, 2006.
13. P. Collet, E. Lutton, M. Schoenauer, J. Louchet, Take it EASEA, in: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature PPSN VI*, Vol. 1917 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2000, pp. 891–901.
14. O. Maitre, F. Kruger, S. Querry, N. Lachiche, P. Collet, EASEA: specification and execution of evolutionary algorithms on GPGPU, *Soft Computing* 16 (2) (2012) 261–279.